

03/8049-SNY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2002-122978  
(P2002-122978A)

(43) 公開日 平成14年4月26日 (2002.4.26)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームト (参考)
G 0 3 F 1/08		G 0 3 F 1/08	A 2 H 0 9 5
G 0 6 F 17/50	6 6 6	G 0 6 F 17/50	6 6 6 C 5 B 0 4 6

審査請求 未請求 請求項の数 4 O L (全 6 頁)

(21) 出願番号 特願2000-317332(P2000-317332)

(22) 出願日 平成12年10月18日 (2000.10.18)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川 6 丁目 7 番 35 号

(72) 発明者 芦田 勲

東京都品川区北品川 6 丁目 7 番 35 号 ソニー株式会社内

(72) 発明者 小川 和久

東京都品川区北品川 6 丁目 7 番 35 号 ソニー株式会社内

(74) 代理人 100086298

弁理士 船橋 國則

Fターム (参考) 2H095 BB01

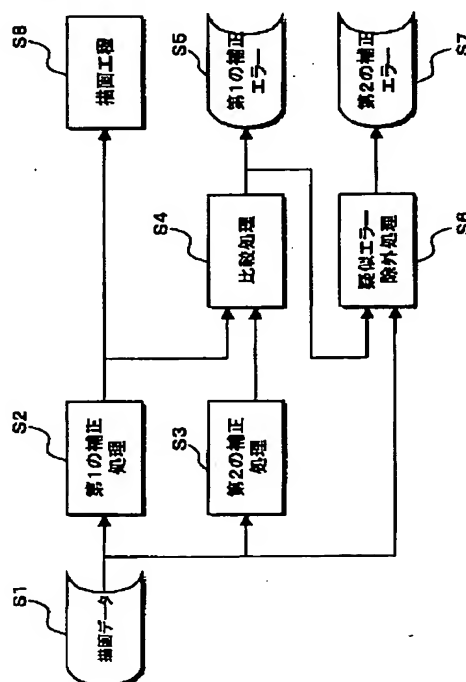
5B046 AA08 BA04 FA04 FA05

(54) 【発明の名称】 マスクデータの検証方法および検証プログラムを記録したコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 マスクデータの補正をプログラムで行うにあたり、プログラムのエラー等を的確に検証すること。

【解決手段】 本発明のマスクデータの検証方法は、マスクデータについて所定条件に基づく補正を行うため、アルゴリズムの異なる複数のプログラムによって各々補正後のマスクデータを作成する工程 (S2、S3) と、作成された補正後の各マスクデータを比較する工程 (S4) と、比較の結果、補正後の各マスクデータに相違がある場合、その相違からマスクデータとして問題となるものを抽出する工程 (S6) とを備えている。



## 【特許請求の範囲】

【請求項1】 マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、  
前記作成された補正後の各マスクデータを比較する工程と、  
前記比較の結果、補正後の各マスクデータに相違がある場合、その相違からマスクデータとして問題となるものを抽出する工程とを備えることを特徴とするマスクデータの検証方法。

【請求項2】 マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、  
前記作成された補正後の各マスクデータを比較する工程と、  
前記比較の結果、補正後の各マスクデータに複数の相違がある場合、その相違からマスクデータとして問題とならない部分を取り除き、残った相違からマスクデータとして問題となるものを抽出する工程とを備えることを特徴とするマスクデータの検証方法。

【請求項3】 マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、  
前記作成された補正後の各マスクデータを比較する工程と、  
前記比較の結果、補正後の各マスクデータに相違がある場合、その相違からマスクデータとして問題となるものを抽出する工程とを備えることを特徴とするマスクデータの検証プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項4】 マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、  
前記作成された補正後の各マスクデータを比較する工程と、  
前記比較の結果、補正後の各マスクデータに複数の相違がある場合、その相違からマスクデータとして問題とならない部分を取り除き、残った相違からマスクデータとして問題となるものを抽出する工程とを備えることを特徴とするマスクデータの検証プログラムを記録したコンピュータ読み取り可能な記録媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、半導体装置等の製造工程で用いられるマスクのパターンデータが適正であるか否かを検出するマスクデータの検証方法および検証プログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

## 【0002】

【従来の技術】半導体装置の製造に用いられるマスクパ

ターンを作成するにあたり、ウェハ製造余裕度向上等を目的として、光近接効果補正を代表とするパターン形状変化を伴う補正処理を用いることが多くなってきている。

【0003】この中で、例えばルールベースの1次元光近接効果補正（Optical Proximityeffect Correction：以下、「OPC」と言う。）の場合、例えば、表1に示されるような補正条件を与えることで、図7に示すような形で補正処理を実施する方法がある。

## 【0004】

【表1】

		補正対象スペース			
		a	b	c	...
補正対象線幅	A	$\alpha$	$\beta$	$\gamma$	...
	B	$\delta$	$\epsilon$	$\zeta$	...
	C	$\eta$	$\theta$	$\iota$	...
	⋮	⋮	⋮	⋮	⋮

【0005】表1および図7に示す例の場合、補正の対象となるマスクパターンの線幅それぞれについて、隣接パターンまでの距離との関係で補正量を決定している。図7に示すマスクパターンP1では、線幅がB、隣接パターンP2までの距離がaであることから、表1に示す補正量として $\delta$ が得られ、この結果、対象となるマスクパターンP1の線幅Bを $\delta$ だけ太らせる補正を行う。

【0006】このようにマスクパターンの補正量を決定し、補正を実施する作業は、一般的に計算機上のソフトウェアにより実現されている。

【0007】ところで、このソフトウェアによる補正量の計算過程において、計算機の突発的なエラー（メモリーエラー等）や、アルゴリズムに基づく計算ミスなど、何らかの原因により補正処理が正常に行われなかった場合、正常でないマスクパターンが作成されてしまい、半導体装置等の製造上重大な問題を与えることになる。そこで、マスクパターンの補正が正しく行われているか否かを検証する必要がある。

## 【0008】

【発明が解決しようとする課題】しかしながら、このようなマスクパターンの補正処理は、パターン形状の変化を伴うため、それが正常に行われたか否かを容易に検証することは難しい。ここで、従来、マスクパターンの検証という観点から、種々の技術が開示されている。

【0009】例えば、パターン上でいくつかの確認ポイントを設定しておき、その部分を何らかの手段で人為的に測長し、その結果が補正条件に対して正しい場合、全体が正しいものとする検証方法がある。ところが、この検証方法では、補正条件の取り違えなど、パターン全体

に影響を及ぼすトラブルの検出については有効であるものの、それ以外のトラブルの場合、確認ポイントに偶然トラブルが発生しない限り検証することはできない。

【0010】また、特開平11-174659号公報に開示されているように、使用している補正テーブルに示されている最大補正量を超えたパターン変化があるか否かを確認する検証方法もある。この検証方法では、補正前後のパターンに対し、比較的単純な図形演算処理で実現できるものの、発生している問題によるパターン異常が、最大補正量を超えていない場合は検証不可能である。

【0011】さらに、特開平11-184064号公報、特開平11-282151号公報に開示されているように、補正結果のパターンがリソグラフィ工程において問題なく解像するかをシミュレーションで確認する検証方法もある。しかし、この検証方法は、補正結果のパターンがリソグラフィ工程で問題を起こさないことを保証するものであり、そのマスクパターンが設計上意図したものであるか否かの保証にはならない。しかも、シミュレーションによる検証には非常に多くの時間を要するという問題がある。

【0012】

【課題を解決するための手段】本発明はこのような課題を解決するために成されたものである。すなわち、本発明のマスクデータの検証方法は、マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、作成された補正後の各マスクデータを比較する工程と、比較の結果、補正後の各マスクデータに相違がある場合、その相違からマスクデータとして問題となるものを抽出する工程とを備えている。

【0013】また、マスクデータについて各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成する工程と、作成された補正後の各マスクデータを比較する工程と、比較の結果、補正後の各マスクデータに複数の相違がある場合、その相違からマスクデータとして問題とならない部分を取り除き、残った相違からマスクデータとして問題となるものを抽出する工程とを備えるマスクデータの検証方法でもある。

【0014】このような本発明では、マスクデータについて所定条件に基づく補正を行うにあたり、各々異なるアルゴリズムから構成される複数のプログラムによって各々補正後のマスクデータを作成している。ここで、各プログラムは各々異なるアルゴリズムから構成されるものの、目的は同じなので同じ補正条件であれば同じ補正後のマスクデータを出力することになる。このことから、各プログラムで作成した補正後の各マスクデータを比較して、相違があった場合、その相違からマスクデータとして問題となるものを抽出することでマスクデータ

の的確な検証を行うことができる。

【0015】

【発明の実施の形態】以下、本発明の実施の形態を図に基づいて説明する。図1は、本実施形態に係るマスクデータの検証方法を説明する図である。すなわち、本実施形態は、マスクデータによって記述されるマスクパターンについて、光近接効果補正(OPC)を代表とするパターン形状変化を伴う補正処理を施すにあたり、その補正後のマスクパターンが正常に補正されているか否かを検証するものである。

【0016】特に、本実施形態では、マスクデータについて光近接効果補正等の所定条件に基づく補正を行う場合、各々異なるアルゴリズムから構成される複数のプログラム(例えば、2つのプログラム)によって各々補正後のマスクデータを作成し、これら補正後のマスクデータを比較して、その比較結果に基づきマスクデータの検証を行う点に特徴がある。

【0017】すなわち、本実施形態におけるマスクデータの検証方法は、マスクの描画データを作成し(S1)、第1の補正処理として第1のプログラムによるOPC処理を実行し、第1の補正結果を算出する(S2)。

【0018】また、第1の補正処理と平行して、同じ描画データを入力とした第2の補正処理を行い、第2の補正結果を算出する(S3)。この第2の補正処理は、第1のプログラムと目的は同じでもアルゴリズムが異なるプログラムを用いる。

【0019】次に、第1の補正結果と第2の補正結果との比較処理を行う(S4)。この比較の結果、第1の補正結果と第2の補正結果とに相違があれば、これを第1の補正エラーとして出力する(S5)。この第1の補正エラーには、後の工程(フォトリソグラフィ工程)で問題になるエラー(相違点)と、問題にならないエラーとの両方が含まれる。

【0020】そこで、この第1の補正エラーから疑似エラーを取り除く疑似エラー除去処理を行う(S6)。ここで、疑似エラーとは、第1の補正エラーのうち後の工程で問題にならないエラーのことである。この疑似エラー除去処理を行うことで、後の工程で問題となるエラーのみが第2の補正エラーとして出力されることになる(S7)。第2の補正エラーが出力された場合には、第1の補正処理もしくは第2の補正処理のうち、いずれかに補正エラーがあるため、補正エラーのない方の補正結果を用いて描画工程(S8)へと移行する。

【0021】このような本実施形態の検証方法における基本的な考え方は、同じ目的のプログラムであっても別のアルゴリズムを持つものであれば、両者が同じ誤りを起こす可能性は極めて低いということに基づいている。つまり、両プログラムが正常に処理されていれば、仕様が明示する範囲内での結果(同じ条件に基づく結果)は

一致するはずである。

【0022】ところが、プログラムの仕様で不明確な部分があると、その不明確な部分において両プログラムの処理結果に相違が発生する可能性がある。

【0023】例えば、表1に示すような線幅のマスクパターンについての補正を目的とする場合、このようにマスクパターンの線幅が明確に示される部分については異なるアルゴリズムのプログラムであっても同じ結果が得られることになるが、マスクパターンが複雑になると線幅の特定が難しくなり、各プログラムにおいてアルゴリズムの違いによる補正結果の相違が現れる可能性がある。

【0024】図2、図3は、複雑なマスクパターンの一例を説明する模式図である。このうち図2は、線幅 $W_1$ と $W_2$ とから成るL字型のマスクパターンPaの例を示している。このようなマスクパターンPaでは、図中斜線で示す部分において、どちらの線幅 $W_1$ 、 $W_2$ を用いて補正すればよいか規定されておらず、どのように処理するかは各プログラムのアルゴリズムに任せられている。

【0025】また、図3に示すマスクパターンPbでは、線幅 $W_3$ 、 $W_4$ 、 $W_5$ が組み合わされた構成となっており、図中斜線で示す部分では、どの線幅 $W_3$ 、 $W_4$ 、 $W_5$ を用いて補正すればよいか規定されておらず、どのように処理するかは各プログラムのアルゴリズムに任せられている。

【0026】図4は、図2に示すマスクパターンの補正例を示す模式図で、(a)は第1のプログラムによる補正例、(b)は第2のプログラムによる補正例を示している。いずれも図中二点鎖線によって示される部分が補正部分である。

【0027】図4(a)に示すように、第1のプログラムでは、マスクパターンPaの偶部(図中破線部分)についても他の部分と同じように補正が施されている。一方、図4(b)に示すように、第2のプログラムでは、マスクパターンPaの偶部(図中破線部分)には補正が施されていない。これは、両プログラムのアルゴリズムの違いによるもので、仕様が不明確な部分は独自の考え方にに基づきアルゴリズムを構成しているためである。

【0028】また、図5は、図3に示すマスクパターンの補正例を示す模式図で、(a)は第1のプログラムによる補正例、(b)は第2のプログラムによる補正例を示している。いずれも図中二点鎖線によって示される部分が補正部分である。

【0029】図5(a)に示すように、第1のプログラムでは、マスクパターンPbの屈曲部(図中破線部分)についても他の部分と同じように補正が施されている。一方、図5(b)に示すように、第2のプログラムでは、マスクパターンPbの屈曲部外側(図中破線部分)には補正が施されていない。これは、先と同様に、両プログラムのアルゴリズムの違いによるもので、仕様が不

明確な部分は独自の考え方にに基づきアルゴリズムを構成しているためである。

【0030】また、図6は、マスクパターンの端部の補正例を示す模式図で、(a)は第1のプログラムによる補正例、(b)は第2のプログラムによる補正例を示している。いずれも図中二点鎖線によって示される部分が補正部分である。

【0031】図6(a)に示すように、第1のプログラムでは、マスクパターンPcの端部についても他の部分と同じように補正が施されている。一方、図6(b)に示すように、第2のプログラムでは、マスクパターンPcの端部には補正が施されていない。これも先と同様に、両プログラムのアルゴリズムの違いによるもので、仕様が不明確な部分は独自の考え方にに基づきアルゴリズムを構成しているためである。

【0032】このようにアルゴリズムの異なるプログラムでは、補正結果に相違が現れる場合があり、図1に示す第1の補正エラーでは、単純に第1のプログラムによる第1の補正結果と、第2のプログラムによる第2の補正結果との相違が示されることになる。

【0033】本実施形態では、この第1の補正エラーから疑似エラーを除外する処理を行い、後の工程で問題となる補正エラーのみを抽出している。ここで、疑似エラーの除外処理としては、図4から図6で示した両プログラムのアルゴリズムの違いによる補正結果の相違を疑似エラーとする場合と、疑似エラーとしない場合とが考えられる。

【0034】つまり、図4から図6に示した両プログラムのアルゴリズムの違いによる補正結果の相違は、仕様として明確になっていない部分であり、本来、後の工程で問題とならない部分であることが多い。他のマスクパターンレイアウトから、このような考え方ができる場合には、図4から図6で示す補正結果の相違点を疑似エラーとして、疑似エラー除外処理によって第1の補正エラーから取り除くようにする。

【0035】一方、図4から図6に示す補正結果の相違点が他のマスクパターンレイアウト等の関係から後の工程で問題となる場合には、この相違点を疑似エラーにはせず、疑似エラー除外処理で取り除かないようにする。

【0036】この疑似エラーは、その条件を特定することができれば、設計工程で利用されている汎用的ソフトウェアであるDRCによって容易に取り除くことができる。第1の補正エラーから疑似エラーを取り除くことで、プログラム本来のバグや、計算機の突発的な問題で発生した後工程に影響するエラーのみを第2の補正エラーとして抽出できるようになる。

【0037】ここで、疑似エラーは、両プログラムのアルゴリズムの関係によって異なり、プログラムを変えた場合にはそのプログラムのアルゴリズムの相違によって生じる疑似エラーを新たに登録することになる。これに

より、プログラムによる補正結果の誤りを的確に検出することができるようになる。

【0038】また、本実施形態では、アルゴリズムの異なる複数のプログラムを用いて複数の補正結果を生成しているが、図1に示す第1の補正処理(S2)および第2の補正処理(S3)のようにこれらを並列処理することで、計算時間の遅延を防ぐことができる。

【0039】さらに、本実施形態では、図1に示す第1の補正処理(S2)、第2の補正処理(S3)、比較処理(S4)、疑似エラー除外処理(S6)などをソフトウェアとして実現し、CD-ROM等の媒体に格納してもよい。すなわち、このCD-ROMから本実施形態を構成するプログラムを計算機へインストールすることで、本実施形態を所望の計算機上で実現できるようになる。

【0040】また、上記説明した実施形態では、マスクパターンの補正結果を生成するため主として2つのプログラムを用いたが、本発明はこれに限定されず3つ以上のプログラムを用いても可能である。また、マスクパターンの補正としてOPC処理を対象としたが、これ以外の補正処理であっても適用可能である。

【0041】

【発明の効果】以上説明したように、本発明によれば次のような効果がある。すなわち、複数のプログラムによ

る補正後のマスクデータの相違から、マスクデータとして問題となるものを的確に抽出することができ、マスクデータの正確な検証を行うことが可能となる。また、プログラムのアルゴリズムによるエラーや、計算機の突発的な問題によるエラーを見つけることができ、このエラーによる後工程への悪影響を確実に防ぐことが可能となる。

【図面の簡単な説明】

【図1】本実施形態に係るマスクデータの検証方法を説明する図である。

【図2】複雑なマスクパターンの一例を説明する模式図(その1)である。

【図3】複雑なマスクパターンの一例を説明する模式図(その2)である。

【図4】図2に示すマスクパターンの補正例を示す模式図である。

【図5】図3に示すマスクパターンの補正例を示す模式図である。

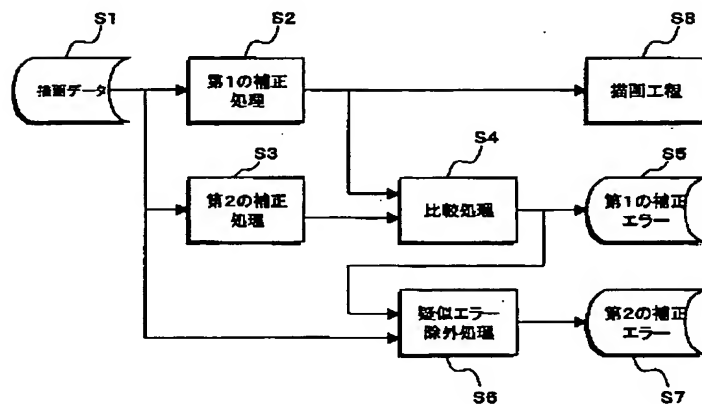
【図6】マスクパターンの端部の補正例を示す模式図である。

【図7】マスクパターンの補正例を示す模式図である。

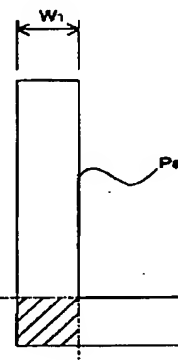
【符号の説明】

S2…第1の補正処理、S3…第2の補正処理、S4…比較処理、S6…疑似エラー除外処理

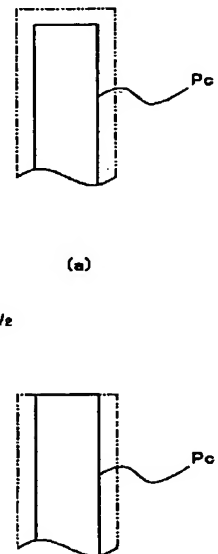
【図1】



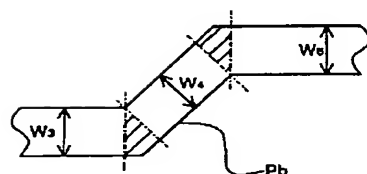
【図2】



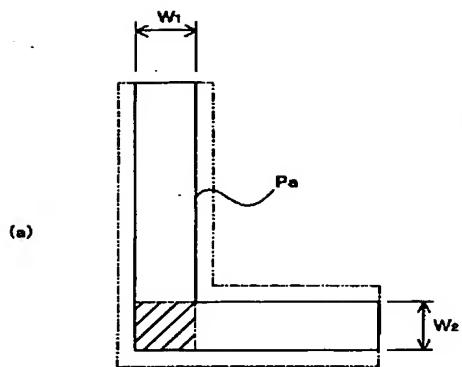
【図6】



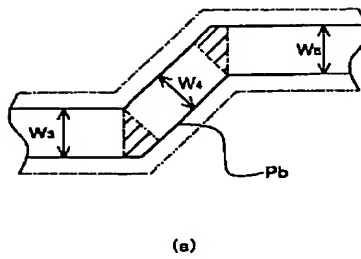
【図3】



【図4】



【図5】



【図7】

